VALIDATING PROGRAMS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of priority to European application serial no. 02022042.2, filed Oct. 1, 2002.

BACKGROUND

[0002] The present invention relates to data processing by digital computer, and more particularly to the validation of computer program applications.

[0003] A computer program application can include a computer program implemented in a compiler language, e.g., an object oriented programming language. Definition modules and implementation modules can be used to describe a computer program implemented in a compiler language. The implementation modules and definition modules can be stored in files of distinct types. In some object oriented programming languages, the definition modules are called interfaces and the implementation modules are called classes.

[0004] The two-component, definition-implementation model aids program verification by allowing early detection of some programming errors. The definition modules or interfaces can be viewed as a record of promises given by a class (the provider class). This record can be used to detect certain programming errors, e.g., where a second class (the customer class) relies on or attempts to use features of the provider class that have not been promised by the provider class. Interface modules can also be used to verify that the implementation of a provider class provides all the promised features.

[0005] Some computer program applications include script code sections. Script code sections can be implemented using an interpreted or scripting language, e.g., JavaScript or Perl. For purposes of this specification, a scripting language is a language that does not support interfaces and that is either weakly typed or untyped. Scripting languages can interact with other programs or with a set of functions provided by an interpreter for the scripting language, as with the file system functions provided in a UNIX shell. Because scripting languages do not support the definition-implementation model, it is difficult to detect errors in script code sections before runtime.

SUMMARY OF THE INVENTION

[0006] In general, in one aspect, the invention provides methods and apparatus, including computer program products, for the validation of computer program applications. The techniques include receiving the language-independent description of a computer program, while editing the language-independent description, generating a language-dependent program from the language-independent description, and validating the language-dependent program. The language-independent description includes a definition module and an implementation module. The language-dependent program includes an interface and a class.

[0007] Advantageous implementations of the invention include one or more of the following features. Validating the language-independent description can include validating the syntax of the definition module and the implementation module. Validating the language-dependent program can include compiling the interface and the class. The definition module and the implementation module can be represented in a meta-language or using a tree structure.

[0008] In another aspect, the invention provides methods and apparatus implementing techniques for the validation of programs including receiving a language-independent description of a computer program, validating the language-independent description, generating a language-dependent program from the language-independent description, and validating the language-dependent program. The language-independent description can include a definition module and an implementation module. The language-dependent program can include a script code section.

[0009] Advantageous implementations of the invention include one or more of the following features. Validating the language-dependent program can include extracting language elements from the script code section, and comparing the extracted language elements with the definition module. Extracting language elements can include generating a symbol table from the script code section. Generating the language-dependent program can include generating language-dependent code comprising an interface and a class. Validating the language-dependent program can include extracting language elements from the script code section, comparing the extracted language elements with the definition module, generating language-dependent code comprising an interface and a class, and compiling the interface and the class.

[0010] In another aspect, the invention provides methods and apparatus implementing techniques for validating programs. The techniques include receiving a language-independent description of a computer program, where the language-independent description includes a definition module and an implementation module, and validating the language-independent description. A first language-dependent program is generated from the language-independent description where the first language-dependent program includes a first script code section. A second language-dependent program is generated from the language-dependent description, where the second language-dependent program includes a second script code section of a distinct second kind. A first set of language elements is extracted from the first script code section, a second set of language elements is extracted from the second script code section, and the first set of language elements and the second set of language elements are compared with the definition module.

[0011] The invention can be implemented to realize one or more of the following advantages. Computer applications can be validated independently of the specific language used to implement the applications. Computer applications can also be validated at an early stage in the validation cycle. Computer applications implemented using programming languages that do not support the concept of validation or the concept of classes and interfaces, can be validated.

[0012] The details of one or more implementations of the invention are set forth in the accompanying drawings and the description below. Other features, aspects, and advantages of the invention will become apparent from the description, the drawings, and the claims.